Universidad Nacional de Asunción
Lic. en Ciencias Informáticas
Facultad Politécnica

Análisis de Algoritmos

Prof. Ing. Derlis Zárate
ProfDerlisZarate@gmail.com

Algoritmos como tecnología

Eficiencia de un algoritmo

- ¿Una máquina más rápida o un algoritmo más eficiente?

Algoritmos como tecnología

Eficiencia de un algoritmo

- Suponga un problema grande.
- Existen varios algoritmos que lo resuelven, pero nosotros elegiremos dos para comparar: el algoritmo A y el algoritmo B.
- Ambos son algoritmos correctos.
- Se sabe que el Algoritmo A (AA) necesita ejecutar N*N instrucciones para resolver el problema de tamaño N, mientras que el Algoritmo B (AB) necesita ejecutar N log N instrucciones para el mismo tamaño

Algoritmos como tecnología

Eficiencia de un algoritmo

- Suponga que tiene dos máquinas X e Y
  - X puede ejecutar 1.000.000.000 instrucciones por segundo
  - Y puede ejecutar 10.000.000 instrucciones por segundo
  - c1 es 2 y c2 == 50 (constantes multiplicativas para AA y AB respectivamente). ¿Qué significa esto?
- Se quiere procesar 1 millón de datos (N=1.000.000)... y la pregunta es ***¿quién ganará?***

Algoritmos como tecnología

Eficiencia de un algoritmo

- El algoritmo A toma alrededor de $c1.n^2$ en tiempo para procesar n items.
- El algoritmo B toma alrededor de c2.n lg n en tiempo (lg es log base 2).
- c1 y c2 son factores constantes.

Asignaremos el AA en la máquina X y el AB en la máquina Y para responder a la pregunta principal: ***Máquina más rápida o Algoritmo más eficiente***.

Algoritmos como tecnología

Algoritmo A en la máquina X

$$\frac{2.(10^6)^2 \, instrucciones}{10^9 \, instrucciones / segundo} = 2000 \, segundos$$

Algoritmo B en la máquina Y

$$\frac{50.(10^6) \, lg \, 10^6 \, instrucciones}{10^7 \, instrucciones / segundo} \approx 100 \, segundos$$

¿Qué pasa cuando n = diez millones?
¿Con cuál algoritmo se quedaría?
¿Cuál es el efecto de la rapidez de la máquina?

Algoritmos como tecnología

¿Cómo escojo el mejor algoritmo?

- Benchmarking: Medir el tiempo de respuesta, pero dependo...
  - ...del subconjunto de pruebas escogidas
  - ...del ordenador donde hago las pruebas
  - ...del lenguaje de programación utilizado
  - ...del compilador utilizado
  - ...de la calidad de la programación
  - Etc.

Para hacer esto tengo que codificar/programar y en ocasiones esto no es trivial.

Algoritmos como tecnología

¿Cómo escojo el mejor algoritmo?

- *Otra alternativa:*
  - Predecir el comportamiento del algoritmo sin necesidad de implementarlo.
    - Análisis Matemático de Algoritmos.
    - Análisis Asintótico de Algoritmos.

Análisis de Algoritmos

- El análisis de algoritmos estudia, desde el punto de vista teórico, los recursos computacionales que necesita la ejecución de un programa de computadora: su eficiencia.
- Es un técnica de estimación. Es útil sin embargo para determinar si vale o no la pena implementar un algoritmo en particular.

Análisis Matemático de Algoritmos

- Consiste en tratar de asociar con cada algoritmo una función matemática exacta que mida su “eficiencia”.
- Para ésto, dependemos de:
  - Las características estructurales del algoritmo.
  - El tamaño del conjunto de datos de entrada.

- Se define una función T(*n*) como la cantidad de trabajo realizado por el algoritmo para procesar una entrada de tamaño *n* y producir una solución al problema.

Calcular el tiempo de ejecución de un algoritmo :: T(n)

Algunas consideraciones sobre las operaciones

- Una asignación = 1 OE (Operación elemental)
- Una comparación = 1 OE
- Acceso a arreglo = 1 OE
- Llamada a función = 1 OE
- Consideramos todas las operaciones del mismo costo (aunque no es real)

Cálculo del tiempo de ejecución de un algoritmo

```
sum = 0                              // 1

for ( i=1; i <= n ; i++ )    // 1 + n * ( 1 + 1 )

   sum++;                            // n * 1
```

```
------------
t (n) = (1) + (1 + 2n) + (n)

t (n) = 2 + 3n

t (n) = 3n + 2   // fijarse que se coloca en termino
                // de operaciones elementales.
```

Análisis Matemático de Algoritmos

- El Análisis Matemático de Algoritmos es independiente de la implementación, sin embargo tiene varios inconvenientes:
  - La imposibilidad de determinar, para muchos problemas y cada una de las entradas, la cantidad de trabajo realizado.
  - La dificultad de determinar T(*n*) exactamente.

Análisis Asintótico de Algoritmos

Tipos de análisis

- Peor caso (usual)
  - Tiempo máximo necesario para un problema de tamaño N
- Caso medio
  - Tiempo esperado para un problema de tamaño N. Implica estudio estadístico.
- Mejor caso
  - Es el más engañoso.

Análisis Asintótico

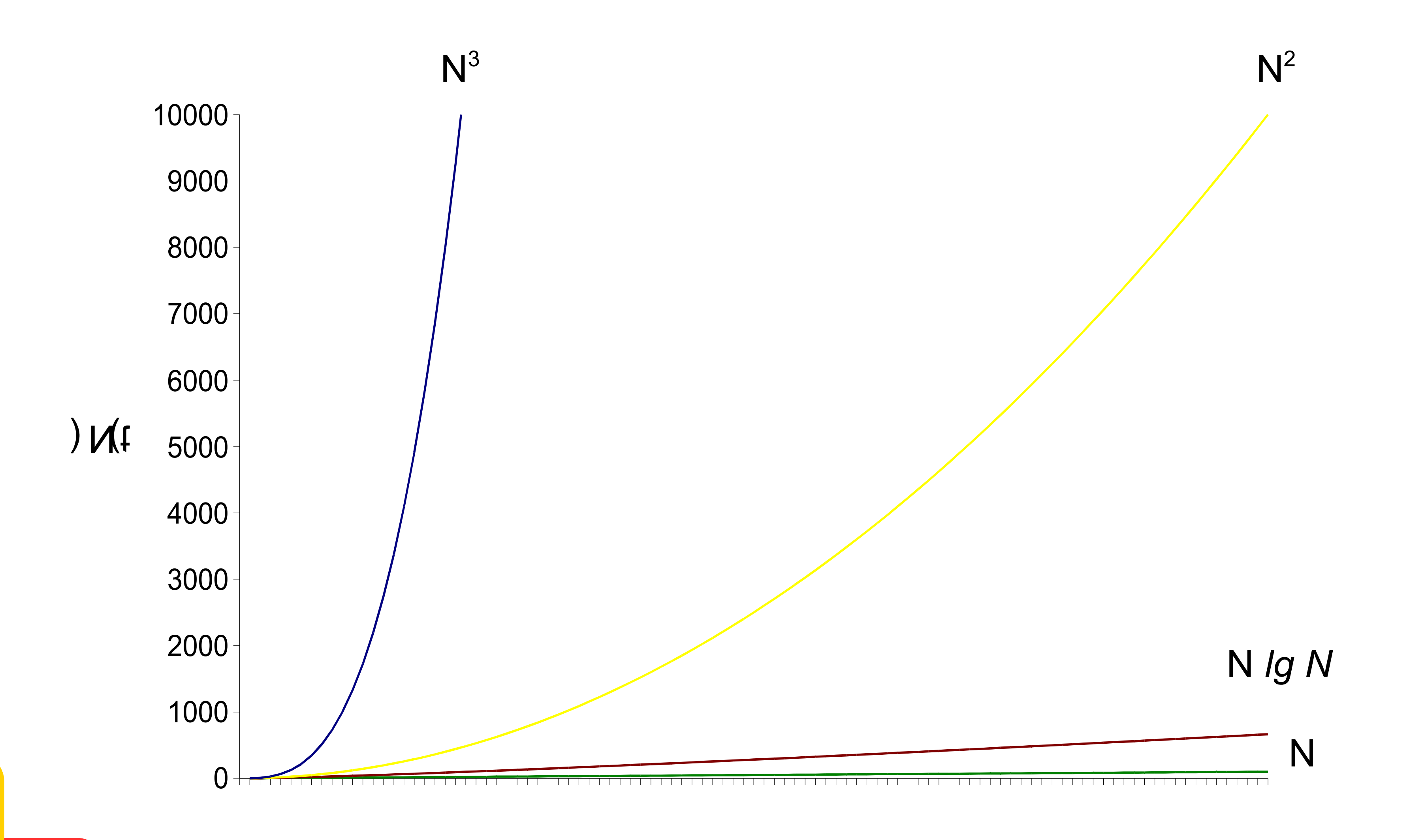
- El análisis asintótico de un algoritmo me permite clasificarlo de acuerdo a su “tasa de crecimiento” (growth rate).
- Ignoramos las constantes que puedan tenerse y nos concentramos en el grado dominante. De ésta forma puedo comparar algoritmos de una forma más sencilla.

Funciones en orden creciente

Función	Nombre
C	Constante
log N	Logarítmica
$\log^2 N$	Logarítmica al cuadrado
N	Lineal
N log N	N log N
N^2	Cuadrática
N^3	Cúbica
2^N	Exponencial

Crecimiento

Crecimiento de algunas funciones



Tamaño de la entrada (N)

Notación O-grande

Definición: Sea ***g*** : N → R* una función arbitraria de los números naturales a los número reales positivos. **O(*g*)** representa al conjunto de todas las funciones ***T*** : N → R si existe una constante real positiva ***M*** y un número natural n_0 tales que para todo $n > n_0$ se tiene que $T(n) <= M*g(n)$.

Notación asintótica

Notación O (O grande) $T(n) <= M * g(n)$.

Notación Ω (Omega) $T(n) >= M * g(n)$.

Notación Θ (theta) $T(n) = M * g(n)$.

Notación o (o chica) $T(n) < M * g(n)$.

Comprobación de Análisis

- Para comprobar empíricamente si el análisis realizado es correcto, se puede implementar el programa y ver si su tiempo de ejecución observado coincide con la predicción
- Para ello, se puede calcular de la siguiente manera: Para determinar si un programa es O(F(N)), se calcula T(N)/F(N) para una serie de valores de N.
- Si F(N) es una respuesta ajustada al tiempo de ejecución, entonces los valores calculados convergen a una constante positiva.
- Si F(N) es una sobreestimación, los valores calculados convergen a cero.
- Si F(N) es una subestimación, los valores divergen

Referencias

- Estructuras de Datos en Java. Mark Allen Weiss, Capítulo 5
- A Practical Introduction to Data Structures and Algorithm Analysis. Shaffer, Cap. 3
- Algoritmos y Estructuras de Datos. Alfred Aho. Cap. 9

Gracias por su Atención

¿Consultas?